

## Examen Partiel

**Exercice 1** Donnez la complexité  $\Theta(\cdot)$  de l'algorithme suivant en fonction de l'entier  $n$  en paramètre en précisant les constantes  $c_1$ ,  $c_2$  et  $n_0$  que vous avez utilisées.

```
public static void algo(double x, int n){
    for(int i=0;i<n;++i)
        for(int j=i;j<n;++j)
            x *= j;
}
```

**Exercice 2** Soit  $T(n)$  le temps d'exécution d'un algorithme en fonction de  $n$  satisfaisant  $T(n) = 8T(n/2) + f(n)$ . Donnez la valeur asymptotique  $\Theta(\cdot)$  de  $T(n)$  dans les trois cas suivants: (vous appliquerez le théorème du cours en précisant la valeur de  $\varepsilon$ )

- 1)  $f(n) = 4n^2 + 5 \lg n$ ;
- 2)  $f(n) = n^3 + 8n^2$ ;
- 3)  $f(n) = \frac{n^4}{10}$ ; (vous appliquerez le corollaire en donnant d'abord une constante  $k$  et un  $n_0$  tels que  $f(2n) \leq kf(n)$  pour tout  $n \geq n_0$ ).

**Exercice 3** Que font les fonctions `xxxx` et `yyyy`?

```
public static double xxxx(int n, int p)
{ if(p>n) return 0.0;
  double res = 1.0;
  for(int i = n ; i >= p ; i--) res *= i;
  return res;
}

public static double yyyy(int n, int p)
{ if(p>n) return 0.0;
  if(p==n)
    return (double) n;
  else
    return yyyy(n, ((n+p)/2)+1) * yyyy((n+p)/2,p);
}
```

- 1) Donnez leur complexité  $\Theta(\cdot)$  en fonction de  $m := n - p$ .
- 2) Démontrer par induction sur  $i$  que `xxxx` fait bien ce que vous avez dit qu'elle faisait.
- 3) Démontrer par induction sur  $m$  que `yyyy` fait bien ce que vous avez dit qu'elle faisait.

**Exercice 4** On considère la complexité de la fonction `main` ci-dessous en fonction du paramètre d'entrée  $n$ :

```
import java.util.Random;
public class Multi{
    public static int increm(int n){if(n==0) return 0; else return ++n;}
    public static int decrem(int n){if(n==0) return 0; else return --n;}
    public static int MultiDecrem(int n, int k){
        while(n != 0 && k !=0){
            n=decrem(n);
            k--;
        }
        return n;
    }
    public static void main (String...args){
        int n = args[0];
        final int seed = args.length > 1 ?
            Integer.parseInt(args[1]) : (int) System.currentTimeMillis();
        final Random random = new Random(seed);
        int compt=n;
        while(compt--!=0){
            int alea = random.nextInt(3);
            if(alea==0) n=increm(n);
            if(alea==1) n=decrem(n);
            if(alea==2) n=MultiDecrem(n,n/2);
        }
    }
}
```

0) Après avoir taper sur un terminal la commande `javac Multi.java`, quelle est ensuite la différence entre `java Multi 50 544534` et `java Multi 50`?

- 1) Montrer que la complexité est  $\Omega(n)$  et  $O(n^2)$ .
- 2) Montrer que la complexité est  $\Theta(n)$ .